AEROSPACE REPORT NO.
TOR-2008(8550)-8033

Software Technology Readiness Assessments—Managing Technology Risks in Space System Acquisitions

16 June 2008

Dr. Peter Hantos
Software Acquisition and Process Department
Software Engineering Subdivision

Prepared for:

Space and Missile Systems Center
Air Force Space Command
483 N. Aviation Blvd.
El Segundo, CA 90245-2808

Contract No. FA8802-04-C-0001

Authorized by: Engineering and Technology Group

Trademarks

AEROSPACE REPORT NO.
TOR-2008(8550)-8033

Software Technology Readiness Assessments—Managing Technology Risks in Space System Acquisitions

16 June 2008

Dr. Peter Hantos
Software Acquisition and Process Department
Software Engineering Subdivision

Prepared for:

Space and Missile Systems Center
Air Force Space Command
483 N. Aviation Blvd.
El Segundo, CA 90245-2808

Contract No. FA8802-04-C-0001

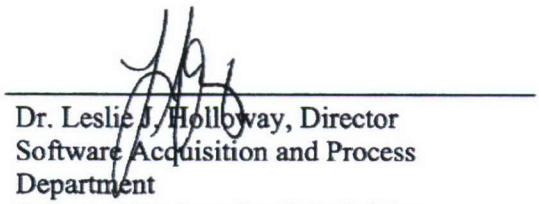
Authorized by: Engineering and Technology Group

THE AEROSPACE CORPORATION

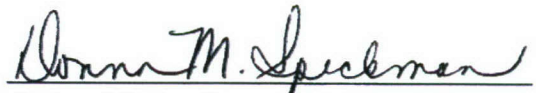
AEROSPACE REPORT NO.
TOR-2008(8550)-8033

Software Technology Readiness Assessments—Managing Technology Risks in Space System Acquisitions

Approved by:



Dr. Leslie J. Holloway, Director
Software Acquisition and Process Department
Software Engineering Subdivision
Computers and Software Division
Engineering and Technology Group



Donna M. Speckman, Director
Research and Program Development Office
Engineering and Technology Group

Abstract

Technology Readiness Assessment (TRA) is an important factor in the milestone decisions of the defense acquisition system. The Department of Defense (DOD) offers substantial direction on assessing hardware technologies in its TRA Deskbook. However, with respect to software, particularly software used in space systems, the guidance has proven to be weak and ambiguous. The objective of this report is to offer tangible guidance on establishing Technology Readiness Levels (TRLs) for space software and provide further insights into all dimensions of technology risk mitigation. To illustrate key points, a comparison of technology risk management within commercial, market-driven companies versus currently codified risk management methods of the defense acquisition community is provided.

Acknowledgements

I want to thank my colleagues on The Aerospace Corporation's team led by Nick Sramek for the support in and for the opportunity to work with them on technology readiness issues.

Contents

Figures

1. Introduction

Technology is central to developing state-of-the-art weapon systems. At the same time, the defense acquisition system wrestles with assessing technology readiness across programs. The 2006 Defense Acquisition Performance Assessment (DAPA) Report [DAPA 2006] summarized the situation as follows: "The inability to define and thus measure technology readiness facilitates decisions to incorporate immature technology in system design at Milestone B which subsequently leads to technical problems during System Design and Development." DOD Instruction (DODI) 5000.2 mandates an independent TRA at all milestones, and it is a particular source of tension that for Milestone B, where the system acquisition program actually starts, it states that the required maturity level is TRL 6 or higher. On the one hand, contractors and the Acquisition Program Offices are told what specific TRL is required for passing the milestone; on the other hand, substantial ambiguity surrounds the determination of TRLs. The DAPA conclusion is particularly troubling considering the fact that a several-hundred-page deskbook is available from the DOD describing the TRA rules [DOD 2005]. Unfortunately, an analysis of the deskbook reveals that it is inadequate when it comes to space system acquisitions, and it is particularly ambiguous with respect to software assessments. This report takes on a double challenge: defining a feasible methodology to assess Software Technology Readiness (STR) for National Security Space (NSS) system acquisitions.

2. Technology Readiness Assessments in Commercial, Market-driven Companies

The following definitions of Technology and Technology Readiness are generic enough to be applicable to defense acquisitions as well, but since they originated from commercial sources, they are briefly described here.

What is Technology? Technology is the practical application of scientific knowledge in a particular domain or in a particular manner to accomplish a task [Foreman 1997].

What is Technology Readiness? The following definition and the overview of a commercial approach for assessing Technology Readiness are based on Xerox Corporation's experiences [Hantos 1998]. Technology Readiness is a state of understanding from which a product can be designed and implemented with predictable performance, costs, delivery, and quality characteristics. Technology Readiness Level is the measure of technology maturity. High TRLs indicate a high level of confidence in that no special solutions would have to be invented beyond normal design engineering practices to satisfy the planned product's requirements. In statistical process control terms, high hardware TRL implies that the centerline of the design capability is centered on manufacturing variability and the distribution of manufacturing variability remains within the design latitude. Hardware Technology Readiness is demonstrated by meeting five criteria:

- Failure modes are identified
- Control Parameters that control these failure modes have been defined
- Safe operating latitudes for failure modes and their controlling parameters are optimized
- Manufacturability requirements are met
- Hardware is capable of delivering at the required performance levels in the absence of identified failure modes

Manufacturing Readiness, distinct from Manufacturability as specified in criterion #4, is also an important risk identification mechanism, but beyond the scope of this paper. Given that software is not manufactured, Software Technology Readiness assessment diverges from the hardware approach. Additionally, the human element (one subtle interpretation of the word "soft" in software) is more prevalent than in hardware development and manufacturing. The referenced Xerox approach to Software Technology Readiness is based on the "Whole Product" concept, i.e., a "Whole Product" includes the core technology and the availability of a range of adjunct products, services, and capabilities [Moore 1991]. Consequently, STR is based on three main components of technology delivery (a variation of the well-known triad): Software Domain Knowledge, Process, and People. In fact, in the absence of "Software Manufacturing Readiness," the People and Process dimensions also encompass the development-organization-chartered technology implementation. In other words, the Xerox assessment approach includes not only a view of the "past," i.e., the evaluation of technology development efforts leading to the assessment, but also a view into the "future," looking at selected dimensions of the upcoming challenges as well.

Commercial Mode of Operation. In commercial, market-driven companies such as General Electric, Hewlett-Packard, IBM, Xerox, etc. there is a pronounced organizational separation between their research centers and development divisions, and the TRA functions as a gating mechanism between the two. These companies can afford to initiate product development more conservatively, mature the new technologies longer in their research centers, and experiment with breakthrough technologies in only a few products. More recently, though, these companies have been trying a new approach—to move several products rapidly through the development pipeline, even if some of them fail at market. As Tom Peters described this situation, instead of "Ready, Aim, Aim, Aim … the new battle call is Fire, Fire, Fire!" [Peters 2002]. The strategic challenge these large companies face is to determine the proper balance in their product portfolio while keeping the product development pipeline saturated. One reason this approach is favored by executive management is that it seems to have a stimulating effect on innovation.

Benefits and Risks Associated with Using New Technology. With respect to risks, commercial companies might not understand their customers well; they might be unsure of the extent of customer interest and the price the market will bear. In the case of hardware products, the manufacturing costs, manufacturing yields, and ramp-up details may not be fully comprehended as well. However, the benefits of using new technology are clear: the opportunity to open new markets and the ability to leverage the technology across many products. Commercial behavior typically involves a Technology Push. Until product marketing commences, many potential customers are not even aware of the new features (e.g., Apple's iPhone). In addition, a segment of the customer base might be so-called "early adopter" technology junkies who enjoy experimenting with cutting-edge products.

The previous analysis is useful because there is a natural desire to adopt commercial approaches, but in defining risks and benefits for the government, the approach is different. In defense acquisition, we are in a Technology Pull mode: we know that the product capabilities required can be only delivered with specific, cutting-edge technologies. In addition, experimentation with launched weaponry is not only undesirable but also unacceptable. Leveraging technology across a defense portfolio is a noble goal, but current policies and inherent acquisition mechanisms actually restrain and in some cases prevent it.

3. The Technology Readiness Assessment Process in Defense Acquisition

The following definitions apply when assessing technology readiness as it relates to defense acquisitions.

What is the Definition of Technology Maturity? Technology maturity is a measure of the degree to which proposed critical technologies meet program objectives. As such, technology maturity is a principal element of program risk [DOD 2005].

What is a Technology Readiness Assessment? A TRA is a systematic, metrics-based process and accompanying report that assesses the maturity of certain technologies used in systems. A TRA examines program concepts, technology requirements, and demonstrated technology capabilities in order to determine technology maturity. The TRA is not intended to predict future performance of the evaluated technologies, nor does it assess the quality of the system architecture, design, or integration plan [DOD 2005].

Overview of the TRA Process. The following steps represent a process that is carried out in the context of an IPA (Independent Program Assessment) by an independent team and are repeated at every major milestone of the DODI 5000.2 process. The first step is the identification of Critical Technology Elements (CTEs) by the Program Manager. A technology element is "critical" if the system being acquired depends on this technology element to meet operational requirements, and if the technology element or its application is either new or novel. With respect to identifying CTEs, the TRA Deskbook recommends the use of the WBS (Work Breakdown Structure) or the System Architecture. The next step is to determine the TRL for all CTEs using a 9-level scale provided by the Deskbook. The final step is to provide information to the Milestone Decision Authority (MDA).

4. Issues in Assessing Software Technology Readiness for (NSS)

When assessing software technology readiness for NSS, issues fall into three categories: (1) military acquisitions-related, (2) space, and (3) software.

4.1 Military Acquisition-related Issues

TRL ratings tend to be viewed as absolute. However, the spirit and the actual text of appropriate DOD policy documents suggest that TRL ratings are in fact relative. TRAs are always conducted in the context of a particular acquisition associated with a well-defined mission (see the previously quoted DOD definition of technology maturity.) This contextual element has a serious bearing on how details of the rating scheme are interpreted, and also on how TRL numbers should be used.

The TRL scheme is meant to be used—intentionally—differently from the well-known, conventional risk management practices, even though the stated goal is to address technology risks. The TRL scheme is a single, absolute scale, and DODI 5000.2 associates a clear go/no-go decision with the ratings (vs. the conventional risk management approach where both likelihood and impact metrics are provided and the decision about the corresponding risk mitigation action is not driven simply by policy). The defense TRA/TRL approach certainly makes the MDA's task easier, but at the same time, it creates new problems and additional ambiguity. For most people in the trenches of the acquisition process, it is difficult to make the distinction and decide which technology-related risks should be treated with the TRA/TRL approach,

and which risks should be handled in the context of conventional risk reduction and risk management activities. This ambiguity represents a particular concern with regard to software.

It is commonly understood and accepted that the methods of determining a TRL are different for hardware and software. However, while hardware approaches are generally understood and thought to be effective, there is wide discomfort with the DOD Deskbook's guidance on TRL schemes for software.

CTE selection is a critical step in the TRA process. A particular difficulty during CTE selection is to decide which domain's (hardware or software) rules and methods should be used for the assessment. For example, let us consider a new digital filter algorithm that eventually might be implemented in either hardware or software. The algorithm itself is a "technology". It is new, and it is critical for achieving mission objectives. At the time of the first TRA, however, most likely we will not have a decision yet about implementation. It must be assessed, but it is unclear which domain's rules should apply. Alternatively, in situations where the solution might call for a nontrivial hardware/software mix, the implementation threads need to be separated and assessed independently in their own domains.

There is also some confusion stemming from the lack of understanding of the NSS acquisition environment. For example, after some extensive critique from the Government Accountability Office (GAO), recommendations were made to use government science and technology laboratories to perform the initial technology development work for space acquisitions [Singer 2005]. However, most NSS satellite acquisitions rely on highly sophisticated technologies, and the development and maturation of those technologies require resources that only large defense contractors have. This fact has major consequences when actual acquisition strategies are determined and contractors engaged.

4.2 Space issues

The DOD Deskbook does not provide specific guidance on assessing space systems. Unfortunately, the common sentiment is that the adopted rating scheme developed by the National Aeronautics and Space Administration (NASA) should be used (see Appendix A). However, the NASA business model is drastically different from the way space systems are acquired for NSS, and this difference becomes crucial to the most discriminating part of the rating scheme. DODI 5000.2 requires at least a TRL-6 for any CTE to pass Milestone B. NASA's requirement for TRL-6 is a system or subsystem model or prototype demonstration in a relevant environment, where the relevant environment for a spacecraft is space. While NASA spacecraft routinely carry technology experiment payloads, Congress is not willing to fund experimentation in NSS acquisition programs. Consequently, relevant environment should be defined in such a way that appropriate verification could be carried out without actually launching the prototypes into space. According to this definition, the relevant environment for space is dissected into four categories (space, launch, designed environment, and operations.) Category details represent the relevant environmental components, such as radiation, vibration, heat, etc. During the TRA process, CTEs are analyzed for all categories of relevant environment. To avoid the need for a technology demonstration in space, appropriate verification methods are specified that cover not only the individual environmental components but account for their cumulative impact as well. The "Designed Environment" category is so named because it covers elements under the control of the designer, while the factors that characterize all the other categories are a-priori given. The dissection of relevant environment is important for software, because neither the space nor the launch environmental categories have direct

impact on the maturity of software technologies, and it would be both prohibitive and unnecessary to require advance space testing for software technologies.

4.3 Software Issues

It is universally agreed that software is a risky business. Why this is the case is not universally understood. First, there is a philosophical conundrum related to the human dimension. Software offers seemingly unlimited opportunities to rapidly leverage and exploit human knowledge. However, the presence of people brings in human factors that are hard to comprehend and even harder to control. The ubiquity of the human factor has lead to the desire to strengthen process and even to entertaining the idea of "software factories." However, in some instances these efforts may impede or even strangle innovation. Second, software in large systems has complexity and other systemic concerns that are difficult to quantify, are usually transparent, and continuously grow during development. Third, even if the software is built mostly with COTS (Commercial Off-The-Shelf) components that are deemed highly reliable building elements), the complexity and systemic concerns remain. Unfortunately, they may become more serious since we have limited visibility to the architecture of the COTS products. Ironically, COTS and other sorts of reuse efforts bring in new type of risks and challenges.

4.3.1 What is Software Technology?

Software technology is a very broad term, and is not defined in the DOD materials. The following discussion is based on [Foreman 1997]. Software technology is defined as the theory and practice of various sciences (including computer, cognitive, statistical sciences, and others) applied to software development, operation, understanding, and maintenance. Specifically, software technology is any concept, process, method, algorithm, or tool whose primary purpose is the development, operation, and maintenance of software or software-intensive systems. Technology includes not just the technical artifacts, but the knowledge embedded in those artifacts and the knowledge required for their effective use. Software technology may include the following:

- Technology directly used in operational systems, such as two tier/three tier software architectures, public key digital signatures, RPCs (Remote Procedure Calls), and rule-based intrusion detection.

- Technology used in tools that produce (or help to produce) or maintain operational systems, such as Graphical User Interface (GUI) builders, cyclomatic complexity analyzers, Ada 95 programming language, just to mention a few.

- Process technologies that make people more effective in producing and maintaining operational systems and tools by structuring development approaches or enabling analysis of systems/product lines. Examples include: Personal Software Process (PSP), Cleanroom Software Engineering, and Domain Engineering and Domain Analysis.

4.3.2 Algorithms

The assessment of algorithms is a particularly contentious issue. According to the SEI definition, algorithms qualify as technology, since they represent a practical application of scientific knowledge. However, their implementation can take place either in hardware or in software. Unfortunately, this implementation distinction is not always clear-cut, and the assessment of algorithms in software-intensive systems becomes a special challenge. For example, in the F22 fighter plane 80% (!) of the functionality is

implemented in software. In the satellite business we go so far as to claim: "The software ***is*** the CONOPS." Since software development is about coding algorithms, it is essential to have clear guidance on how to assess algorithms.

First, we need to reiterate that the evaluation of algorithms only makes sense in the context of a specific acquisition, and using a scheme that is promising a generic or absolute assessment of technology maturity is futile. In the case of the previously mentioned digital filter, the first step is to establish the basic feasibility of the algorithm, with the understanding that true feasibility and viability can only be evaluated after an implementation decision has been made and specific program constraints are provided. After the establishment of basic feasibility, the evaluation might continue with either hardware or software technology readiness assessment schemes, depending on the implementation decision. The basic assessment needs to be application-domain-specific and not implementation-domain-specific; e.g., for the digital filter the domain might be signal conditioning, communication, antennae design, etc. However, either manufacturing considerations for hardware implementation, or software development considerations for software implementation, would be needed to provide a complete TRL.

4.3.3 The Separation of Product and Process Technologies

It is a key concern for software that the DOD guidance related to the separation and treatment of product and process technologies is ambiguous at least, and unsatisfactory at most. In defense acquisition there is an intentional separation of software development risks from technology risks. TRAs are not intended to address the capabilities of the acquiring developer organizations, nor do they attempt to assess processes being applied during development [Gold 2005]. In the current system, contractor capability evaluations that are carried out during source selection are supposed to cover process aspects of software development risks. It is an unfortunate fact though that these evaluations, even the new, CMMI®/SCAMPI℠-based ones, are inadequate and still have serious shortfalls [Eslinger 2007].

4.3.4 Using the WBS

Traditionally, software only shows up at the lower levels of the WBS, and those levels are not available at early stages of the acquisition. There are recommendations to raise the profile and visibility of the software by creating a WBS Level-2 Software Common Element [Eslinger 2006], but none of the current NSS acquisitions has implemented this recommendation yet. Consequently, the DOD Deskbook's recommendation to use the WBS for finding CTEs is not very helpful.

4.3.5 Early Positioning of the TRA

Another, similar concern is related to the positioning of the first TRA too early in the acquisition life cycle. At the entry into Milestone B (or its NSS equivalent, KDP-B), only a system architecture exists, but the determination of software CTEs requires that a substantial software effort be in place. It is understood that having a viable and feasible software architecture is key in determining software technology choices, particularly if the technology solutions directly relate to architecture (e.g., client-server, two-tier, SOA, etc.) However, as far as Aerospace-supported acquisitions are concerned, neither the CDC (Concept Design Center)-provided government architecture nor the contract's WBS appear adequate for identifying and assessing most of the software CTEs. As a result, there is a tendency to ignore software during the first TRA, which is particularly troublesome since this TRA at KDP-B supports the formal approval and funding decision for the acquisition.

4.3.6 Miscellaneous Software Concerns Related to the DOD/NASA Schemes

Relevant environment references (beginning from TRL 5) are ambiguous and do not reflect reality with respect to software development environments and tools. In addition, the overall assessment scheme—particularly the higher-than-TRL 5 definitions—do not reflect the realities of the NSS acquisition process either. Last but not least, TRL 6, while it has the most controversial and debated role during TRA, is inadequately defined.

5. NSS Software TRA/TRL Proposal

5.1 Relevant Environment Considerations for Space

The Relevant Environment for software can be reduced to what is called the Designed Environment (see 4.2 above). The following description has been developed to assess STR for the space segment; some details might be slightly different for other (ground, user, etc.) segments.

Software applications "live" in their Designed Environment, which has both hardware and software elements (see Figure 1). Besides their intrinsic characteristics, TRLs for tools and system software depend on the TRL of the hardware platform; TRLs for applications depend on the TRLs of both the hardware platform and the associated tools and system software.

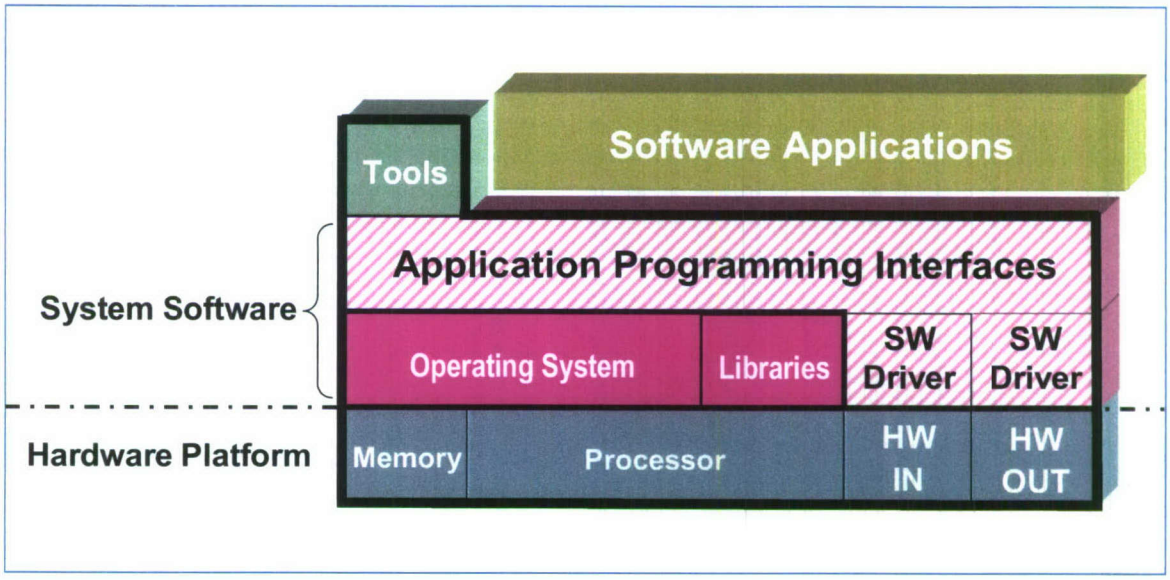


Figure 1. Software Designed Environment

5.2 Determination of Software CTE Candidates

For the determination of CTE candidates, the following three principles are used:

- Principle #1: All software elements of the software designed environment are automatically software CTE candidates. This principle forces the evaluation of technology readiness for the hardware platform elements.
- Principle #2: For application software to be written, first consider if any new or novel software technology solution is needed. If yes, then this technology solution is a software technology CTE candidate. Nevertheless, even if no new technology solutions are needed, prudent practice dictates the use of conventional risk management practices.
- Principle #3: For COTS or reuse software, first consider whether the prior application took place in the same Designed Environment. If yes, it is not a CTE candidate, but traditional trade study justifications and conventional risk management practices still apply. If no, it is automatically a software CTE candidate.

The reason for this selection process is as follows. It is clearly understood that any need for downstream change in either hardware platform, tools, or the system software might put software development in jeopardy, requiring extraordinary efforts that would definitely go beyond "normal" engineering practices. However, this process makes the involved elements only candidates. The true impact and likelihood of a potential change must be assessed on a case-by-case basis.

5.3 Software TRLs for NSS

The proposal is built on the NASA scheme but it is customized to the NSS acquisition environment (see Appendix B). An important part of the new scheme is the way software designed environments are characterized:

- ***Experimental*** environments do not have any constraints; they could reside on any platform, might include analytical tools, simulators, a cross-platform development environment, etc.
- ***Operational-like*** environment means that all elements of the Software Designed Environment at the development site are the same as the planned mission. Note that cross-platform tools are not acceptable.
- ***Actual*** environment means that all elements of the Software Designed Environment at the actual customer site are the same as the planned mission.
- ***Space*** is not really a designed environment; it is associated with completed missions only.

With respect to the TRL definitions, the following changes have been made:

- Level 1-3 details from an acquisition perspective are almost irrelevant. In fact, it would make the scheme cleaner if they were collapsed into one category. However, they are kept separate to ensure consistency with the current system.
- Level 4 has been redefined to better support the new Level 6 definition.
- Level 5 represents a critical milestone in technology development. To achieve this level of maturity the work has to move out of the experimental state and the appropriate objectives have to be accomplished in an "operational-like" environment.
- Level 6 definition now clearly states that the maturity of the CTE in question must be determined in the actual mission's context.

- Level 7 definition now spells out that only segment integration and qualification can validate this rating, using "actual" environment.
- Level 8 definition is the logical evolution of Level 7, moving up on the system's WBS.
- Level 9 definition has not changed; this level of maturity can only be achieved after the first launch.

6. Conclusions

TRA is a specialized risk management mechanism to deal with technology risks. The DOD TRA Deskbook, without additional guidance, seems to be inadequate to carry out TRAs for software, and its shortcomings are particularly problematic for TRAs in NSS acquisitions. In search of solutions, commercial approaches were examined, resulting in the unfortunate conclusion that defense acquisition has unique characteristics that distinguish it from commercial product development, drastically limiting the adoption of otherwise proven and effective commercial TRA practices. The proposal detailed in Section 5 represents an attempt to satisfy the fundamental spirit and intent of TRA for defense acquisition, and not to rewrite but only augment the DOD Deskbook. Due to the inherent constraints of the defense acquisition system, though, most likely a drastic re-evaluation would be needed to define a comprehensive and effective approach to deal with technology risks. However, making such a proposal was out of the scope of this paper.

References

DAPA 2006 DAPA Project, March 2006 http://www.acq.osd.mil/dapaproject/

DOD 2005 Department of Defense Technology Readiness Assessment (TRA) Deskbook, May 2005, http://www.akss.dau.mil/darc/darc.html

Eslinger 2006 Eslinger, S., The Position of Software in the Work Breakdown Structure (WBS) for Space Systems, Aerospace Report TR-2006(8550)-3, December 20, 2006

Eslinger 2007 Eslinger, S., Mission-Assurance Driven Processes for Software-IntensiveSystems, Systems & Software Technology Conference, June 18-21, 2007, Tampa, FL

Foreman 1997 Foreman, J., et al., Software Technology Review, CMU/SEI draft, June 1997

Gold 2005 Gold, R., Jakubek, D., Technology Readiness Assessments for IT and IT-enabled Systems, CrossTalk, May 2005

Hantos 1998 Hantos, P., Sie, C., Managing the Transition from Software Technology Development to Product Development, California Software Symposium '98, October 23, 1998, Irvine, CA

Mankins 1995 Mankins, J.M., Technology Readiness Levels – A White Paper, National Aeronautics and Space Administration, 1995, http://www.hq.nasa.gov/office/codeq/trl/trl.pdf

Moore 1991 Moore, G., Crossing the Chasm: Marketing and Selling Technology Products to Mainstream Customers, Harper Business, 1991

Peters 2002 Peters, T., Tom Peters' True Confessions, Fast Company Magazine, Issue 53, pp 78

Singer 2005 Singer, J., GAO Report Calls for Less Technology Development in Military Space Procurement, Space News, February 28, 2005

Appendix A: NASA Technology Readiness Levels*

TRL 1 Basic Principles observed and reported

TRL 2 Technology Concept and/or application formulated

TRL 3 Analytical and experimental critical function and/or characteristic proof-of-concept

TRL 4 Component and/or breadboard validation in laboratory environment

TRL 5 Component and/or breadboard validation in relevant environment

TRL 6 System/subsystem model or prototype demonstration in a relevant environment (ground or space)

TRL 7 System prototype demonstration in space environment

TRL 8 Actual system completed and flight-qualified through test and demonstration (ground or space)

TRL 9 Actual system "flight proven" through successful mission operations

---

* Source: [Mankins 1995]

Appendix B: National Security Space Software Technology Readiness Levels

SW TRL	Definition	Supporting Information	Software Designed Environment
1	Demonstration of basic principles	No constraints on DE	**Experimental**
2	Concept of application formulation	No constraints on DE	
3	Proof of concept	Simulation or prototyping is conducted, no constraints on DE	
4	Fundamental characterization of technology	Critical performance parameters are identified	
5	Characterization in operational-like setting	Dependency of critical performance parameters on the SW DE is identified	**Operational-like**
6	Qualification against applicable TPMs of the actual mission	Operational latitudes for critical parameters established and validated	**Operational-like**
7	Validation via **segment** integration & qualification	(Other segments are ***simulated*** for end-to-end testing)	**Actual**
8	Validation via **system** integration & qualification	Other segments are ***available*** and ***used*** for end-to-end testing)	**Actual**
9	Validation in completed space mission	n/a	**Space**